

Adaptive DNA Computing Algorithm by Using PCR and Restriction Enzyme

Shinpei Watanabe, Yusei Tsuboi, Zuwairie Ibrahim, Tsuneto Yamamoto, Osamu Ono
Department of Electrical and Electronic Engineering
Meiji University
1-1-1 Higashi-mita, Tama-ku, Kawasaki-shi
Kanagawa-ken, 214-8571 JAPAN

Abstract—In this paper, we introduce a new adaptive DNA computing algorithm by using Polymerase Chain Reaction (PCR). The adaptive algorithm is designed based on Adleman-Lipton [3] paradigm of DNA computing. However, unlike the Adleman-Lipton architecture, a cutting operation has been introduced to the algorithm and the mechanism in which the molecules used by computation were feedback to the next cycle was devised. Moreover, amplification by PCR is performed in the molecule used by feedback and a concentration difference arises in the base sequence can be used again. By doing this, the molecules which serve as a solution candidate can be narrowed down and the optimal solution can be detected easily. From the application point of view, a simulation has been carried out on the *shortest path problem* and the validity of the proposed adaptive algorithm is stated from the results of the simulation. Finally, we go on to propose applying adaptive algorithm to the chemistry experiment which used the actual DNA molecules for solving a universal problem.

Keywords—DNA computing; algorithm; Polymerase Chain Reaction; optimization; shortest path problem

I. INTRODUCTION

In 1994, Leonard Adleman [1] successfully solved the Hamiltonian path problem (HPP) of seven nodes using the actual DNA molecules. Following his success, a new research field called DNA computing has been established. However, in DNA computing, there is a problem is that the quantity of the molecules used for computation will increase exponentially with the scale of the problem. In this paper, we introduce an adaptive DNA-based computing algorithm by using Polymerase Chain Reaction (PCR) which is a new method aiming to improve the explosion problem of the DNA molecules.

The adaptive algorithm which will be proposed here is based on the Adleman-Lipton paradigm. The Adleman-Lipton paradigm tends to discover the solution of a problem by a series of biochemical operations. Based on this paradigm, for solving HPP, the DNA molecules encoding all the possible routes are poured into a test tube. After the hybridization and

ligation, based on the Watson-Crick [2] complementarity, it is expected that there will exist a combination among various kind of combinations that is represents the solution to the problem. Thus, before the computation begins, it is necessary to prepare beforehand sufficient DNA molecules in order to generate the wanted combinations. The main disadvantage of this procedure is that the quantity of required DNA molecules increases as the problem size increases.

In order to solve this limitation, we propose an adaptive algorithm which consists of two concepts: study and strengthening. Study is memorizing the knowledge acquired from experience and building new knowledge structure. Acquisition of the knowledge by study enables the suitable response to the given input. The study function in DNA computing introduces the feedback structure of the detection result of a solution. By doing this, the detected molecules can be memorized as knowledge, and the function of knowledge change by storing a new knowledge for the next computation.

The strengthening function in DNA computing is realized by concentration control of DNA molecules. That is to say, by amplifying a molecule using Polymerase Chain Reaction (PCR), the concentration of the molecule is changed dynamically. Consequently, the concentration of the molecules suitable for detection of a solution can be made. By combining a feedback mechanism and dynamic concentration control, adaptive algorithm adjusts the concentration value of the molecules by 1 cycle computation, and reuses the results of concentration adjustment to the next cycles. Therefore, this algorithm does not need to search a solution in a series of operations. Furthermore, it is able to cut down the quantity of DNA molecules by learning in order to increase the accuracy of the computation.

The proposed adaptive algorithm based on PCR is applied to the shortest path problem, and the correctness is verified by performing a computer simulation.

II. THE ADAPTIVE ALGORITHM

The adaptive algorithm consists of six steps as shown in the Figure 1 below.

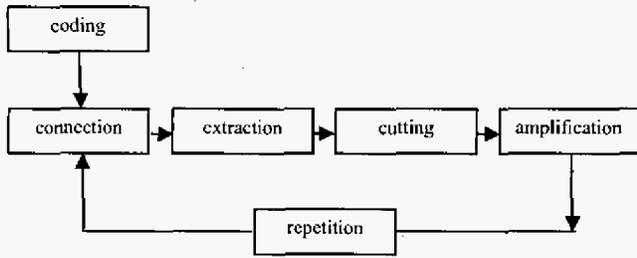


Figure 1. The flow of the adaptive algorithm

- Step 1 (coding): The given problem is encoded by the DNA molecules.
- Step 2 (connection): The coded DNA molecules are made to connect.
- Step 3 (extraction): The combination which is a solution candidate is extracted among the generated combinations.
- Step 4 (cutting): The extracted combination is cut by using a restriction enzyme.
- Step 5 (amplification): The cut combination is amplified by using PCR.
- Step 6 (repetition): Step 2 to step 5 are repeated.

As such, a graph with three vertices and four edges as shown in the Figure 2 is considered. This graph gives as an example, and the procedure of coding is explained in detail. For that reason, especially, this graph is not set up as a problem.

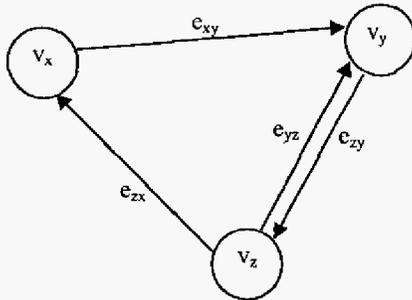


Figure 2. A graph of three vertices and four edges

Firstly, oligonucleotides, or oligos for short, are assigned to the vertex $v_x, v_y,$ and $v_z,$ and edge $e_{xy}, e_{zx}, e_{yz},$ and $e_{zy}.$ The oligos are made of 20 nucleotides in this graph. In order to simplify the notation of base sequence, the sign described below expresses the nucleotides assigned to each vertex.

$v_x:$ GCATTCGGAT CTAGCATCFG	$v_x:v_x(p) v_x(n)$
$v_y:$ TACGGTTCCA GCATTGGGTC	$v_y:v_y(p) v_y(n)$
$v_z:$ AGGGCATTCC CATTCCGAGT	$v_z:v_z(p) v_z(n)$

Each vertex $v_i(p)(i=x,y,z)$ expresses 10 nucleotides in the first half of the oligos which encodes the vertex $v_i,$ and each vertex $v_i(n)(i=x,y,z)$ express 10 nucleotides in the second half of the oligos which encodes the vertex $v_i.$ In the adaptive algorithm, α and β segments are used as recognition part of a restriction enzyme between $v_i(p)$ and $v_i(n)$ as described below. In here, α and β segments are corresponded to a restriction enzyme EcoR I. That is to say, α are assigned AATTC, β are assigned G. Consequently, if α and β segments connect with complementary $\sim\alpha$ and $\sim\beta$ segment, it will be designed so that it may become the recognition part of a restriction enzyme EcoR I.

$$\begin{aligned}
 v_x:v_x(p) &| \alpha \beta | v_x(n) \\
 v_y:v_y(p) &| \alpha \beta | v_y(n) \\
 v_z:v_z(p) &| \alpha \beta | v_z(n)
 \end{aligned}$$

Next, each edge consists of the complementary sequence in the first half of each vertex and in the second half of each vertex. Furthermore, each edge is encoded by adding a restriction enzyme recognition part.

$$\begin{aligned}
 e_{xy}:\sim v_x(n) &| \sim v_y(p) | \sim \alpha \sim \beta \\
 e_{yz}:\sim v_y(n) &| \sim v_z(p) | \sim \alpha \sim \beta \\
 e_{zy}:\sim v_z(n) &| \sim v_y(p) | \sim \alpha \sim \beta \\
 e_{zx}:\sim v_z(n) &| \sim v_x(p) | \sim \alpha \sim \beta
 \end{aligned}$$

where, a sign $[\sim]$ means complementary relation. Each encoded vertices and edges alternately connected by Watson-Crick complementarity. The procedure of coding described above can apply to coding of the shortest path problem.

III. THE SIMULATION TO THE SHORTEST PATH PROBLEM

We applied the proposed adaptive algorithm to the shortest path problem. The search space is identified by denoting the lower left of the space as search start point and the upper left of the space as a goal, in a $n \times n$ square search space as described in Figure 3. An object in the search space is movable in the eight directions of the four-direction slant in the position which does not touch a wall.

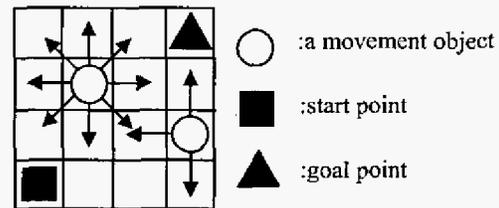


Figure 3. A search space of the shortest path problem

Each space is called domains. In case movable four directions (top, diagonal right, right, diagonal below), the quantity of a DNA molecules required in order to express the number of combinations in order to reach the goal is computed. For example, in the search space of $6 \times 6,$ 4.6×10^4 kinds of all combinations of a goal attainment path exist. If

the number of average the number of times of movements is expressed as the average of the shortest number of times of movement and the longest number of times of movement, the average number of times of movement will become 10 times. If one vertex is expressed by 20 nucleotides, at least $4.6 \times 10^4 \times 20 \times 10 = 9.2 \times 10^6$ nucleotides are required to express all goal attainment paths. Table 1 expresses the molecular weight of four kinds of nucleotides. If the number of nucleotide which exists in 1 gram is computed using the average of four molecular weights, it will become 1.842×10^{21} pieces. Therefore, the needed nucleotides for expressing all goal attainment paths is 5.1×10^{-15} grams. Similarly, the results of the computed quantity of DNA molecules used for other search space are summarized in Table 2.

TABLE 1. MOLECULAR WEIGHT OF NUCLEOTIDE

Nucleotide	pdA	pdG	pdC	pdT
Molecular weight	331.22	347.22	307.20	322.21

TABLE 2. THE AMOUNT OF ESTIMATED DNA MOLECULE

Search space	6×6	8×8	20×20	30×30
Total path	4.6×10^4	6.9×10^6	1.6×10^{20}	2.7×10^{31}
DNA weight	5.1×10^{-15}	1.1×10^{-12}	68.0	1.7×10^{13}

A. Coding and connection

Firstly, the sequence for domain and the sequence for movement as described in Figure 4 are prepared. The sequence for domain is encoded for each domain in the square search space. The sequence for movement is encoded for a path moving from a domain to other domain. $V_i(p)$ expresses nucleotides in the first half of the oligos and $V_i(n)$ expresses nucleotides in the second half of the oligos. α and β express sequence used as recognition part of a restriction enzyme. However, a sign [~] expresses complementary relation. Ligation is performed using all the designed sequence for domain and sequence. Consequently, various combinations are generated. At this time, in the case of the Adleman technique, all combinations are needed to obtain an optimal solution. However, in the case of the proposed algorithm, all combinations are not needed. It is because, even if the optimal solution does not exist in generated combinations, by using feedback structure, the solution candidate is getting close to the optimal solution.

In a computer simulation, if each coding sequence used is in the state of equal concentration, selection of the coding sequence used for connection operation is performed by equal probability, and the connection is performed at random. The

number of times of connection is determined at random for every sequence used for connection and various combinations are generated.



(i) A sequence for domain (ii) A sequence for movement

Figure 4. Coding base sequence

B. Extraction and cutting

The base sequence which serves as a solution candidate among all the combinations generated by connection operation is extracted. There are two conditions for becoming a solution in the shortest path problem. Firstly, it is necessary to have the feature that the combination begins from the domain of a start point, and finishes with the domain of a goal point. Secondly, when the length of the combinations is shorter, it is clear that a possibility which the combinations serve as the optimal solution is high. Therefore, the combination which fulfills these two feature points is extracted, and the combination is considered as superior solution candidate sequence and extracted. Extraction operation is performed by making the amplifying operation using PCR. In a computer simulation, it is enable to control the number of the combinations amplified by using PCR. Therefore, when the fixed combinations are created, PCR is stopped. After that, extracted combinations are cut by the restriction enzyme as described in Figure 5.

C. Amplification and repeated calculation

The cut combinations are detached to single-stranded DNA molecules by denaturation. It is important that one chain of detached the single-stranded DNA molecules serves as the same form as a sequence for movement as described in Figure 6. By supplying a primer described in Figure 7 in the detached sequence, the primer clings to each sequence for movement which has a complementary relation, respectively. And only the coding sequence which constituted the combinations extracted as a solution candidate can be made to amplify by expanding the primer. Each sequence for movement which constitutes the extracted best solution candidate can be amplified. Consequently, a concentration difference arises in each sequence for movement. Each sequence for movement which arise the concentration difference is returned to the process of connection again, and repeated calculation is performed. Under repeated calculation, since a concentration difference arises in each sequence for movement, the high sequence for movement of concentration generates many combinations. Conversely, the low sequence for movement of concentration does not generate many combinations. That is to say, under repeated calculation, generation probability of combinations serves as a solution candidate sequence becomes high compared with before calculation. Thus, by repeating the cycle of connection, extraction, cutting, and

amplification, generation probability of combinations serves as a solution candidate sequence is raised, and finally the adaptive algorithm detect an optimal solution. Although it is

important that how many times repeated calculation is performed, about this, it is verified by actually performing a simulation.

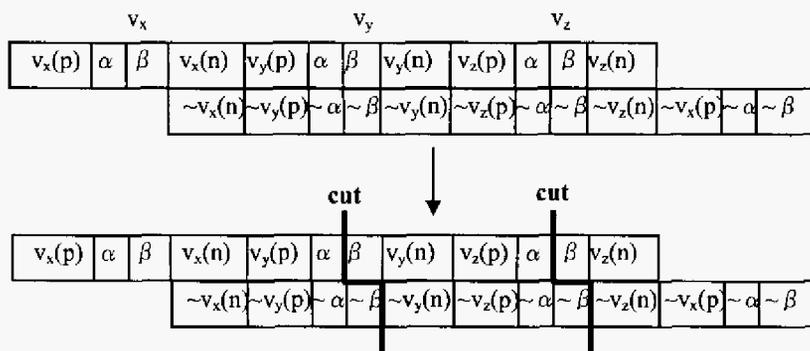


Figure 5. Cutting of the base arrangement

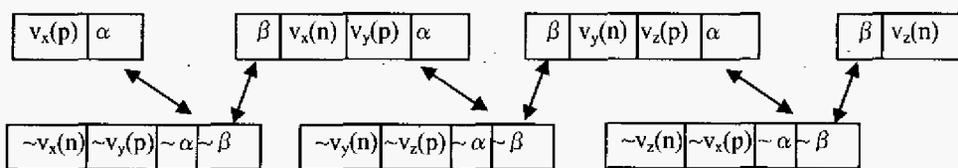


Figure 6. Detachment to single-stranded molecule

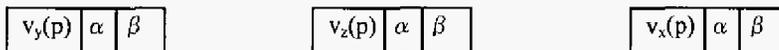


Figure 7. Primer

IV. SIMULATION RESULTS

This section describes the results when applying the Adleman-Lipton paradigm and adaptive algorithm to the shortest path problem.

A. Adleman-Lipton paradigm

The Adleman technique makes all generable path patterns, and detects a solution from the all candidates of solution. However, it is difficult to reproduce all the combinations in a limited memory domain on a computer. Therefore, during the simulation, the number of path was limited beforehand, and was generated, and repetition calculation was performed. The number of times of trial and number of times of detection of the optimal solution were decided for the rate of detection of the optimal solution, and the number of patterns of a molecule required for detection of a solution was presumed. The number of patterns of a molecule required for solution detection based on Adleman-Lipton paradigm is described in Table 3. Since it can move in the eight directions in fact but when the total number of paths described in a table can be moved in the four directions, the total number of paths increases drastically. In a solution candidate's path, many paths which reach the destination through the same vertex repeatedly and which are not solutions clearly also exist. On a computer, it is because the generation probability of such a

path was set up very low. We found that this result shows the limitation connection of a molecule and the limitation is extended in a limited memory domain on a computer in order to show the solution. Since the simulation is performed on a computer, it is unavoidable that an error with a theoretical path pattern arises.

B. Adaptive algorithm

Figure 8 (in the search domain of 36, the number of patterns 30) shows the detection result of a solution of an Adleman-Lipton paradigm and adaptive algorithm. Since there were very few patterns as following by Figure 8, with Adleman technique the solution was once undetectable in 50 times trial. In contrast, with adaptive technique the optimal solution was detected by 28 trials and all subsequent trials have detected the solution. The adaptive technique evaluated the result obtained by each trial by the form of a concentration difference, and uses the detected molecule again. Therefore, we found that it converged on the optimal solution by trying the number of times of fixed. We found that the feature which is not in the Adleman technique to which each trial is carried out independently appeared.

Figure 9 (in the search domain of 64) shows the detection result of a solution of the adaptive algorithm when changing the number of patterns. It can be seen that the number of

times of trial required for convergence of the optimal solution is reducible as the number of patterns increases as graph.

Table 4 describes the relation between the number of times of trial and the number of patterns of a molecule to be used in each search domain. In every search domain, it turns out that the number of patterns of the molecule used as the number of times of trial increases can be reduced.

Figure 9 (in 2 times of the number of the times of trial) compares the rate of molecules used between the adaptive algorithm and the Adleman-Lipton paradigm. In every domain, the adaptive technique has fewer rates molecule used than the Adleman technique. Moreover, this result was obtained as the search domain became large because if a search domain becomes large, generation of a useless path will also increase. Thus, the adaptive technique is able to cut down the generation of such a useless path effectively. However, there is a problem of this adaptive algorithm. There may be a limited repetition of chemistry operation and the increase in the number of times of repetition may cause explosion of calculation time.

TABLE 3. THE NUMBER OF PATTERNS OF A MOLECULE REQUIRED FOR DETECTION OF A SOLUTION

Domain	6×6	7×7	8×8
Pattern	1.3×10^3	3.3×10^3	1.7×10^4
Total path	4.6×10^4	5.6×10^5	6.9×10^6

TABLE 4. THE NUMBER OF PATTERNS OF THE NECESSARY MOLECULE

trial/field	5×5	6×6	7×7	8×8
2	3.7.E+02	5.8.E+02	9.5.E+02	2.0.E+03
6	1.1.E+02	2.6.E+02	4.8.E+02	1.0.E+03
10	3.5.E+01	1.2.E+02	2.4.E+02	5.2.E+02
14	1.1.E+01	5.3.E+01	1.2.E+02	2.7.E+02
18	3.4.E+00	2.4.E+01	6.2.E+01	1.4.E+02
20	1.9.E+00	1.6.E+01	4.4.E+01	9.8.E+01

TABLE 5. THE RATE OF CURTAILMENT OF THE MOLECULE (TWO TRIALS)

Domain	5×5	6×6	7×7	8×8
Adleman-Lipton	7.1×10^2	1.3×10^3	3.3×10^3	1.7×10^4
Adaptive Algorithm	3.7×10^2	5.8×10^2	9.5×10^2	3.2×10^3
Rate[%]	52.1	44.6	28.8	11.8

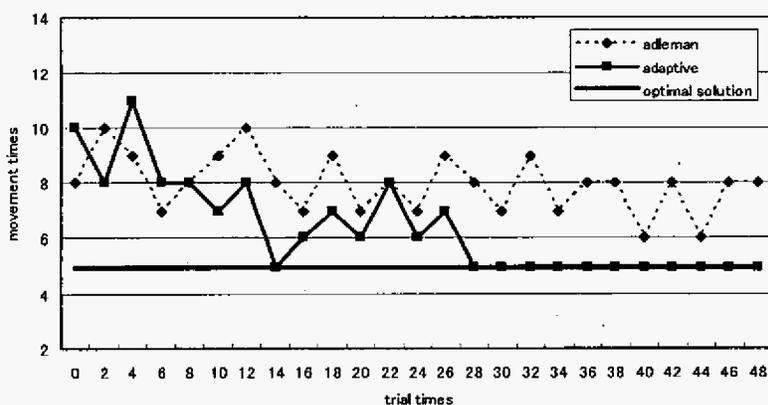


Figure 8. The detection result of a optimal solution (domain:6×6)

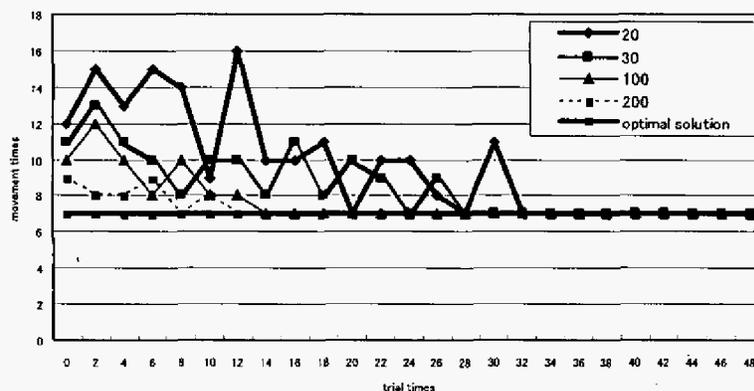


Figure 9. The detection result of a optimal solution (domain:8×8)

V. CONCLUSION

It has been shown in literature that DNA computing provides a lot of potentials. However, there are many critical problems in this research area which must be overcome. One of the problems is the explosion possibility of DNA molecules with respect to the increasing of the size of the problem. In this paper, in order to solve this problem, an algorithm which is based on adaptive DNA molecules by using PCR was introduced. The validity and the correctness of the proposed adaptive algorithm have been proved from the simulation results. Moreover, this adaptive algorithm is able to decrease the generation of useless paths which is caused by the connection of unexpected molecules. It is hope that the proposed adaptive algorithm will provides great enhancements from the molecules explosion point of view for a more applicable DNA-based computing approach.

REFERENCES

- [1] L.Adleman: Molecular Computation of Solutions to Combinatorial Problems, *Science*, Vol.266,1021-1024,1994
- [2] J.D.Watson, F.H.C.Crick: Molecular structure of nucleic acids: a structure for deoxyribose nucleic acid, *Nature* (London), 171,737-738,1953
- [3] R.J.Lipton: DNA Solution of Hard computational Problems, *Science*, Vol.268, 542-545, 1995
- [4] Ogiya Masaki, Yokomori Takatomo, DNA Computer, Baifukan, 2001
- [5] Oouti Higashi, Yamamoto Masato, Kawamura Hidenori, Shiba Keiiti, Takayanagi Toshiaki, Endou Satoshi, Computing Paradigm By Life Complexity System, 82-132,2003
- [6] Sam Roweis, Erik Winfree, Richard Burgoyne, Nickolas V.Chelyapov, Myron F.Goodman, Paul W.K.Rothemund, Leonard M.Adleman: A Sticker-Based Model for DNA Computational Biology, Vol5, 615-629, 1998